



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/940,758	08/28/2001	Jean Louis Calvignac	RAL920000106US1	5788
7590 05/05/2005			EXAMINER	
IBM CORPORATION PO BOX 12195 DEPT 9CCA, BLDG 002 RESEARCH TRIANGLE PARK, NC 27709			NGO, NGUYEN HOANG	
			ART UNIT	PAPER NUMBER
			2663	

DATE MAILED: 05/05/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

## Office Action Summary

**Application No.**

09/940,758

**Applicant(s)**

JEAN LOUIS CALVIGNAC

**Examiner**

Nguyen Ngo

**Art Unit**

2663

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 8/28/2001.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-39 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-39 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 28 January 2002 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- |   |   |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)   | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)  | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)             |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

## **DETAILED ACTION**

### ***Specification***

1. The attempt to incorporate subject matter into this application by reference to "Hash Function in IP, MAC, and Other Structured Addresses" (page9 line 6 and page23 line 1) is improper because applicant has failed to provide the U.S. Patent Application Serial Number or Patent Number.

### ***Claim Objections***

2. The numbering of claims is not in accordance with 37 CFR 1.126 which requires the original numbering of the claims to be preserved throughout the prosecution. When claims are canceled, the remaining claims must not be renumbered. When new claims are presented, they must be numbered consecutively beginning with the number next following the highest numbered claims previously presented (whether entered or not).

Misnumbered claims 23-38 has been renumbered to 24-39. The Examiner believes that there might be a typographical error in claim 22, where –The system as recited in claim 20, – should be considered as a separate claim. Examiner believes this separate claim to be claim 23. Thus appropriate renumbering of claims 23-38 to 24-39 are thus required. In Examinations of claims, the Examiner has considered the above, and all arguments correlate to the new renumbering system (34-39).

The Examiner also believes there might be a typographical error in claims 31-34 and claims 36-39 (based on new renumbering system) as these claims are believed to

Art Unit: 2663

be depended on the wrong claim due to error in numbering. For example the claim below;

**30.** The method as recited in claim **29**, wherein if said search key matches a particular entry in said content addressable memory then said content addressable memory returns said particular entry number.

Should be changed to:

**31.** The method as recited in claim **30**, wherein if said search key matches a particular entry in said content addressable memory then said content addressable memory returns said particular entry number.

Once again, it is noted that the Examiner provides arguments to claims based on the correction of the above.

### ***Claim Rejections - 35 USC § 103***

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.

Art Unit: 2663

2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

5. Claims 1, 12, 30, and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Corl, Jr. et al. (U.S. 6,529,897) in view of Murase (U.S. 2001/0028651) hereinafter referred to as Corl, and Murase, respectively.

**Regarding claim 1**, Corl discloses a method for testing filter rules in which a key for each packet matches and, therefore, an action will be enforced on the packet to which the key corresponds (data structure associated with a packet of data, performing a particular action on said packet of data based on said data structure associated with said packet of data, col5 lines 65-67). Corl further discloses:

an ingress PMM that receive traffic (receiving a packet of data, col5 lines 58).

a key for a packet, which has been obtained from the packet in which the key for a packet may be the IP five-tuple (extracting from packet header of said packet of data to generate a search key, col8 lines 20-23).

a decision tree, which is used for a specific set of filter rules (selecting table to be accessed using said search key, col6 lines 8-9).

that depending on the goals of the system (determining based on a table definition of selected table), searches (identifying said data structure) may be commenced at different times, using the cache or the decision tree (using content addressable memory (CAM) or a tree based, col13 lines 41-46). It is

noted that Examiner will provide further arguments into why cache will correlate to a CAM.

that at a particular step, stored actions (identifying a data structure associated with packet) will be obtained that correspond to the key by using a cache or a decision tree (in response to said determination step, col8 line 34 and lines 42-43).

Corl however is silent in stating the use of a CAM in identifying said data structure associated with said packet.

Murase discloses a cache memory that contains a CAM and a correspondence table (page1 paragraph [0010]) and gives the motivation of having a cache memory, which contains a CAM in order to decrease access time and perform faster searches on a table. It would have thus been obvious to a person skilled in the art to include a cache containing a CAM disclosed by Murase into the method and system for testing filter rules disclosed by Corl in order to speed up key searches.

**Regarding claim 12**, the combination of Corl, and Murase disclose all the limitations as discussed in the rejection of claim 1.

**Regarding claim 30**, Corl discloses a method for testing filter rules in which a key for each packet matches and, therefore, an action will be enforced on the packet to which the key corresponds (data structure associated with a packet of

Art Unit: 2663

data, performing a particular action on said packet of data based on said data structure associated with said packet of data, col5 lines 65-67). Corl further discloses:

an ingress PMM that receive traffic (receiving a packet of data, col5 lines 58).

a key for a packet, which has been obtained from the packet in which the key for a packet may be the IP five-tuple (extracting from packet header of said packet of data to generate a search key, col8 lines 20-23).

that the search of the cache (CAM) starts after the search of the decision tree (transferring search key to a CAM by a tree search, figure 9 and col14 lines 10-14). Corl also discloses that if the key found in the decision tree, then the key is written to cache (transferring search key to a CAM by a tree search, col10 lines 18-19).

searching a plurality of stored keys in the cache for the key (identifying a particular entry number in CAM, col3 lines 63-64). It is further stated that if there is a match (based on search key, whether said search key identifies and matches a particular entry in said CAM), a step in obtaining the stored action(s) corresponding to the key will be taken (identifying said data structure associated with said packet of data based on said particular entry number in said CAM, col8 lines 34).

Corl however is silent in stating the use of a CAM in identifying said data structure associated with said packet.

Murase discloses a cache memory that contains a CAM and a correspondence table (page1 paragraph [0010]) and gives the motivation of having a cache memory, which contains a CAM in order to decrease access time and perform faster searches on a table. It would have thus been obvious to a person skilled in the art to include a cache containing a CAM disclosed by Murase into the method and system for testing filter rules disclosed by Corl in order to speed up key searches.

**Regarding claim 31**, the combination of Corl, and Murase, discloses all the limitations as discussed in the rejection of claims 30. Murase further discloses that a cache table (in said CAM) is searched with a key and if any entry hit is found, the information that a hit entry (search key matches particular entry) was found is notified to the packet processing circuit along with the address of the cache table holding the hit entry (returns a particular entry number, page 3 [0055]).

6. Claims 2, 3, 15, 16, 17, 27, and 32-39 are rejected under 35 U.S.C. 103(a) as being unpatentable over Corl, Jr. et al. (U.S. 6,529,897) in view of Murase (U.S



Art Unit: 2663

2001/0028651) and further in view of Hunter et al (U.S 6,343,289), hereinafter referred to as Corl, Murase, and Hunter, respectively.

**Regarding claim 2**, Corl and Murase discloses all the limitations of claim 1 and it is further shown that Corl discloses the determination step, that the cache contains the hashed key among stored hashed keys (determined to use said CAM to identify data structure, col10 lines 4-6). Corl further discloses that if the hash key has not been found in the cache, then the action for the hashed key is obtained using a search of the decision tree (transferring said search key to a tree search engine, col10 lines 19-20).

The combination of Corl and Murase is however silent in associating a key with a particular thread number/table number pair.

Hunter discloses outputting an address (thread number/table number associated with said search key) to the memory that corresponds to the location in memory storing the desired key (search key, col21 lines 55) and gives the motivation for an improved organization and search of a forwarding table base in order to enhance performance and speed. It would have thus been obvious to a person skilled in the art to associate said search key with a particular thread number disclosed by Hunter with the method and system for testing filter rules using a cache containing a CAM disclosed by Corl and Murase to provide an improved organization in a search and thus improving process times.

**Regarding claim 3**, the combination of Corl, Murase, and Hunter discloses all the limitations of claim 2 and it is further shown that, Hunter discloses a request to load (key is transfer to a particular address in a first register) a first key from memory that is associated with a database entry. The load requests include outputting an address to the memory that corresponds to a location in memory (first register in tree search engine) storing the desired key (particular address in said first register is used to decode said particular tread number associated with said search key, to col2 lines 49-55).

**Regarding claim 15**, Corl discloses all the limitations as discussed in the rejection of claims 1, as claim 15 is simply the device of the method presented in claims 1, but is however silent in specific limitations correlating to claim 15. Corl discloses a network processor (packet processor) for testing filter rules in which a key for each packet matches and, therefore, an action will be enforced on the packet to which the key corresponds (identifying data structure associated with a packet of data, performing a particular action on said packet of data based on said data structure associated with said packet of data and based using CAM or tree based search engine, col5 lines 65-67). Corl further discloses:

an ingress PMM that receive traffic (internal processor comprising circuitry for receiving a packet of data, col5 lines 58).

a key for a packet, which has been obtained from the packet in which the key for a packet may be the IP five-tuple (extracting from packet header of said packet of data to generate a search key, col8 lines 20-23).

a decision tree, which is used for a specific set of filter rules (selecting table to be accessed using said search key, col6 lines 8-9).

that depending on the goals of the system (determining based on a table definition of selected table), searches (identifying said data structure) may be commenced at different times, using the cache or the decision tree (using content addressable memory (CAM) or a tree based, col13 lines 41-46). Corl is however silent in stating the use of a CAM in identifying said data structure associated with said packet. It is noted that Examiner will provide further arguments into why cache will correlate to a CAM below.

that a switch and the network processor (internal processor) is used for testing of filter rules which utilizes a decision tree, particularly a software managed decision tree to test filter rules (tree search engine coupled to said internal processor, col5 lines 60-64 and col6 lines 1-2).

Corl is however silent in disclosing a data structure memory coupled to said search engine. It is noted that Examiner will provide further arguments below.

Corl is also however silent in disclosing said content addressable memory coupled to said tree search engine via an interface unit. It is noted that Examiner will provide further arguments below.

that at a particular step, stored actions (identifying a data structure associated with packet) will be obtained that correspond to the key by using a cache or a decision tree (in response to said internal processor determining whether to identify said data structure using CAM or tree based search engine, col8 line 34 and lines 42-43).

Corl however is silent in stating the use of a CAM in identifying said data structure associated with said packet.

Murase discloses a cache memory that contains a CAM and a correspondence table (page1 paragraph [0010]) and gives the motivation of having a cache memory, which contains a CAM in order to decrease access time and perform faster searches on a table. It would have thus been obvious to a person skilled in the art to include a cache containing a CAM disclosed by Murase into the method and system for testing filter rules disclosed by Corl in order to speed up key searches.

Corl is however silent in disclosing a data structure memory coupled to said search engine. Corl discloses that if a key matches the filter rule, type of actions may be performed on a specific packet (col2 lines 45-47).

Murase discloses a forwarding table that has entries containing all paths needed in routing (specific action, page1 [0015]). Murase thus gives the motivation of storing specific data structure comprising specific actions to perform

on said packet in a data structure memory (forwarding table) as it is necessary to have memory in order to obtain the type of actions disclosed by Corl. It would have thus been obvious to a person skilled in the art to include a data structure memory disclosed by Murase coupled to said tree search engine disclosed by Corl in order to obtain specific action related to specific packets.

The combination of Corl and Murase is however silent in disclosing said content addressable memory coupled to said tree search engine via an interface unit.

Hunter however, discloses a bus interface, which moves packet data between the fabric interface and the I/O interface (col4 lines 55). Hunter thus provides the motivation of using interfaces in order to effectively move packets between elements. It would have thus been obvious to a person skilled in the art to incorporate an interface coupled between said CAM and said tree search engine disclosed by Hunter with the method and system for testing filter rules using a cache containing a CAM disclosed by Corl, Murase, and Hunter to effectively move packets between elements (tree search engine and CAM).

**Regarding claim 16**, the combination of Corl, Murase and Hunter discloses all the limitations of claim 15 and it is further shown that Corl discloses that if the hash key has not been found in the cache, then the action for the hashed key is

obtained using a search of the decision tree (transferring said search key to a tree search engine, col10 lines 19-20).

The combination of Corl and Murase is however silent in associating a key with a particular thread number/table number pair.

Hunter discloses outputting an address (thread number/table number associated with said search key) to the memory that corresponds to the location in memory storing the desired key (search key, col21 lines 55) and gives the motivation for an improved organization and search of a forwarding table base in order to enhance performance and speed. It would have thus been obvious to a person skilled in the art to associate said search key with a particular thread number disclosed by Hunter with the method and system for testing filter rules using a cache containing a CAM disclosed by Corl and Murase to provide an improved organization in a search and thus improving process times.

**Regarding claim 17**, the combination of Corl, Murase, and Hunter discloses all the limitations of claim 16 and it is further shown that, Hunter discloses a request to load (key is transfer to a particular address in a first register) a first key from memory that is associated with a database entry. The load requests include outputting an address to the memory that corresponds to a location in memory (first register in tree search engine) storing the desired key (particular address in said first register is used to decode said particular tread number associated with said search key, to col2 lines 49-55).

**Regarding claim 27**, the combination of Corl, Murase, and Hunter discloses all the limitations of claim 15.

**Regarding claim 32**, the combination of Corl, and Murase, discloses all the limitations as discussed in the rejection of claim 30, but is however silent in disclosing a null pointer when search key does not match a particular entry.

Hunter however discloses of a null pointer (return of a null pointer) if the entry happens to be the last one in the bin (col1 lines 47-49). Hunter further discloses if the last bin in the hash bin (located in CAM) has been reached, then no matching entry exists (perform search and search key does not match particular entry) and the search is complete (col9 lines 7-12). It would have thus been obvious to a person skilled in the art to associate a null pointer disclosed by Hunter into the method and system for testing filter rules using a cache containing a CAM disclosed by Corl and Murase to provide a detection means of when the key search is complete and whether or not a match is found.

**Regarding claim 33** the combination of Corl, Murase, and Hunter disclose all the limitations as discussed in the rejection of claim 32.

**Regarding claim 34**, the combination of Corl, Murase, and Hunter disclose all the limitations as discussed in the rejection of claim 33. Hunter further discloses

entries, which also include a pointer to the next entry in the bin (loading a register with a pointer, wherein said pointer points to said register, col1 lines 45-46). As mentioned with claim 32, said search is complete only when null pointer is reached for the last one in the bin otherwise search is considered incomplete.

**Regarding claim 35**, Corl discloses all the limitations as discussed in the rejection of claims 30, as claim 35 is simply the device of the method presented in claims 30, but is however silent in specific limitations correlating to claim 35. Corl discloses a network processor (packet processor) for testing filter rules in which a key for each packet matches and, therefore, an action will be enforced on the packet to which the key corresponds (identifying data structure associated with a packet of data, performing a particular action on said packet of data based on said data structure associated with said packet of data and based using CAM or tree based search engine, col5 lines 65-67). Corl further discloses:

an ingress PMM that receive traffic (internal processor comprising circuitry for receiving a packet of data, col5 lines 58).

a key for a packet, which has been obtained from the packet in which the key for a packet may be the IP five-tuple (extracting from packet header of said packet of data to generate a search key, col8 lines 20-23).

that a switch and the network processor (internal processor) is used for testing of filter rules which utilizes a decision tree, particularly a software



Art Unit: 2663

managed decision tree to test filter rules (tree search engine coupled to said internal processor, col5 lines 60-64 and col6 lines 1-2).

that the search of the cache (CAM) starts after the search of the decision tree (transferring search key to a CAM by a tree search, figure 9 and col14 lines 10-14). Corl also discloses that if the key found in the decision tree, then the key is written to cache (transferring search key to a CAM by a tree search, col10 lines 18-19). Corl however is silent in stating the use of a CAM in identifying said data structure associated with said packet. It is noted that Examiner will provide further arguments below.

Corl is however silent in disclosing a data structure memory coupled to said search engine. It is noted that Examiner will provide further arguments below.

Corl is however silent in disclosing said content addressable memory coupled to said tree search engine via an interface unit. It is noted that Examiner will provide further arguments below.

searching a plurality of stored keys in the cache for the key (identifying a particular entry number in CAM, col3 lines 63-64). It is further stated that if there is a match (based on search key, whether said search key identifies and matches a particular entry in said CAM), a step in obtaining the stored action(s) corresponding to the key will be taken (identifying said data structure associated with said packet of data based on said particular entry number in said CAM, col8 lines 34).

Corl however is silent in stating the use of a CAM in identifying said data structure associated with said packet.

Murase discloses a cache memory that contains a CAM and a correspondence table (page1 paragraph [0010]) and gives the motivation of having a cache memory, which contains a CAM in order to decrease access time and perform faster searches on a table. It would have thus been obvious to a person skilled in the art to include a cache containing a CAM disclosed by Murase into the method and system for testing filter rules disclosed by Corl in order to speed up key searches.

Corl is however silent in disclosing a data structure memory coupled to said search engine. Corl discloses that if a key matches the filter rule, type of actions may be performed on a specific packet (col2 lines 45-47).

Murase discloses a forwarding table that has entries containing all paths needed in routing (specific action, page1 [0015]). Murase thus gives the motivation of storing specific data structure comprising specific actions to perform on said packet in a data structure memory (forwarding table) as it is necessary to have memory in order to obtain the type of actions disclosed by Corl. It would have thus been obvious to a person skilled in the art to include a data structure memory disclosed by Murase coupled to said tree search engine disclosed by Corl in order to obtain specific action related to specific packets.

The combination of Corl and Murase is however silent in disclosing said content addressable memory coupled to said tree search engine via an interface unit.

Hunter however, discloses a bus interface, which moves packet data between the fabric interface and the I/O interface (col4 lines 55). Hunter thus provides the motivation of using interfaces in order to effectively move packets between elements. It would have thus been obvious to a person skilled in the art to incorporate an interface coupled between said CAM and said tree search engine disclosed by Hunter with the method and system for testing filter rules using a cache containing a CAM disclosed by Corl, Murase, and Hunter to effectively move packets between elements (tree search engine and CAM).

**Regarding claim 36**, the combination of Corl, Murase, and Hunter discloses all the limitations as discussed in the rejection of claims 35. Murase further discloses that a cache table (in said CAM) is searched with a key and if any entry hit is found, the information that a hit entry (search key matches particular entry) was found is notified to the packet processing circuit along with the address of the cache table holding the hit entry (returns a particular entry number, page 3 [0055]).

**Regarding claim 37**, the combination of Corl, Murase, Hunter discloses all the limitations as discussed in the rejection of claim 35, Hunter further discloses of a null pointer (return of a null pointer) if the entry happens to be the last one in the bin (col1 lines 47-49). Hunter further discloses if the last bin in the hash bin (located in CAM) has been reached, then no matching entry exists (perform search and search key does not match particular entry) and the search is complete (col9 lines 7-12). It would have thus been obvious to a person skilled in the art to associate a null pointer disclosed by Hunter into the method and system for testing filter rules using a cache containing a CAM disclosed by Corl and Murase to provide a detection means of when the key search is complete and whether or not a match is found.

**Regarding claim 38** the combination of Corl, Murase, and Hunter disclose all the limitations as discussed in the rejection of claim 37.

**Regarding claim 39**, the combination of Corl, Murase, and Hunter disclose all the limitations as discussed in the rejection of claim 35. Hunter further discloses entries, which also include a pointer to the next entry in the bin (loading a register with a pointer, wherein said pointer points to said register, col1 lines 45-46). As mentioned with claim 32, said search is complete only when null pointer is reached for the last one in the bin otherwise search is considered incomplete.

7. Claims 4-11, 13, 14, 18-26, 28 and 29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Corl, Jr. et al. (U.S. 6,529,897) in view of Murase (U.S. 2001/0028651) and further in view of Hunter et al (U.S. 6,343,289), and further in view of Basso et al (U.S. 2002/0154634), hereinafter referred to as Corl, Murase, Hunter, and Basso, respectively.

**Regarding claim 4**, the combination of Corl, Murase, and Hunter discloses all the limitations of claim 3 but is however silent in transferring a search key from tree search engine to one or more key register in an interface unit, wherein said interface unit interfaces said tree search engine and said CAM. Hunter however, discloses a bus interface, which moves packet data between the fabric interface and the I/O interface (col4 lines 55). Hunter thus provides the motivation of using interfaces in order to effectively move packets between elements.

Basso discloses a device that stores search key in a register in the interface circuit (transferring said search key from said tree search engine to one or more key registers in an interface unit, page 3 paragraph [0048]).

It would have thus been obvious to a person skilled in the art to incorporate an interface comprising a key register disclosed by Basso with the method and system for testing filter rules using a cache containing a CAM and associating a search key with a particular thread number disclosed by Corl, Murase, Hunter, and Basso to effectively move packets between elements (tree search engine and CAM).

**Regarding claim 5**, the combination of Corl, Murase, Hunter, and Basso discloses all the limitations of claim 4, and Hunter further discloses a step in which the key associated with the first entry in the hash bin is requested by outputting (transferring) its address (particular address used to decode said particular thread number associated with said search key) to the memory (to second register, col8 lines 47-49). This address generation thread is then said to further continue with another step (col8 lines 59-60). Hunter is however silent in referring to an interface unit, but with regards to the arguments in claim 4, it would have been obvious to a person skilled in the art to incorporate a second register in an interface unit to hold said particular addresses to effectively move packets between the elements of the tree search engine and CAM.

**Regarding claim 6**, the combination of Corl, Murase, Hunter, and Basso discloses all the limitations of claim 5, and Murase further discloses that the cache table (comprising a CAM) is comprised of a search key register (page 4 [0063]). It is thus obvious to a person skilled in the art that in order for the key register to store keys and search for matches in the cache, transferring of said search key from one of said search key registers in the interface to the cache (CAM) is necessary.

**Regarding claim 7**, the combination of Corl, Murase, Hunter, and Basso discloses all the limitations of claim 6, and Corl further discloses the method comprise searching a plurality of stored keys in the cache for the key (identifying a particular entry number in CAM, col3 lines 63-64). It is further stated that if there is a match (based on search key, whether said search key identifies and matches a particular entry in said CAM), a step in obtaining the stored action(s) corresponding to the key will be taken (identifying said data structure associated with said packet of data based on said particular entry number in said CAM, col8 lines 34).

**Regarding claim 8**, the combination of Corl, Murase, Hunter, and Basso disclose all the limitations as discussed in the rejection of claim 7.

**Regarding claim 9**, the combination of Corl, Murase, Hunter, and Basso disclose all the limitations as discussed in the rejection of claim 8, and Murase further discloses that a cache table (in said CAM) is searched with a key and if any entry hit is found, the information that a hit entry (search key matches particular entry) was found is notified to the packet processing circuit along with the address of the cache table holding the hit entry (returns a particular entry number, page 3 [0055]).

**Regarding claim 10**, the combination of Corl, Murase, Hunter, and Basso disclose all the limitations as discussed in the rejection of claim 9, and Hunter further discloses that a second entry may quickly be determined by simply adding the forwarding database entry length (adding particular entry number to a base address, Examiner believes determination of forwarding database entry length be comprise of determining starting address) to the address of the previous entry (shifting a particular entry number, col7 lines 7-9). Hunter is however silent in storing a result of said particular entry number added to said base address in result register, and reading said result in said result register.

Murase however, further discloses a hit record database (result register), which is updated when a hit occurs in the cache table (CAM, page 2 [0031]). Murase further discloses a status table be located in the hit record database (figure 4), which stores in each entry, the number of that entry (indexed in said result register based on thread number), the hit address in the cache table (particular entry number in CAM), the hit address in the forwarding table (result and pointer used to index into an appropriate entry in said data structure memory to identify said data structure associated with said packet of data, page4 paragraph [0067]). It would have thus been obvious to a person skilled in the art that this status table also store the result of said particular entry number added to said base address disclosed by Hunter with the hit record database disclosed by Murase, as this is simply the means in finding an address in forwarding table (appropriate entry in said data structure memory to identify said data structure



associated with said packet of data) and storing such results in said result register.

**Regarding claim 11**, the combination of Corl, Murase, Hunter, and Basso disclose all the limitations as discussed in the rejection of claim 10.

**Regarding claim 13**, the combination of Corl, Murase, Hunter, and Basso disclose all the limitations as discussed in the rejection of claim 10, Hunter further discloses that each forwarding database entry has the same general format (col10 lines 17-20). It would have thus been obvious to a person skilled in the art to have to have the pointer in a same format when said tree is used to point to a forwarding database.

**Regarding claim 14**, the combination of Corl, Murase, Hunter, and Basso disclose all the limitations as discussed in the rejection of claim 5 and Hunter further discloses a retrieval, which includes causing a pipeline memory (one or more search key registers and said second register organized in pipeline manner) to access memory locations in an order that minimizes a worst case search of the forwarding database (access done by having a search key associated with one or more thread number/table number pairs, col3 lines 9-10).

**Regarding claim 18**, the combination of Corl, Murase, and Hunter discloses all the limitations of claim 17 but is however silent in transferring a search key from tree search engine to one or more key register in an interface unit, wherein said interface unit interfaces said tree search engine and said CAM. Hunter however, discloses a bus interface, which moves packet data between the fabric interface and the I/O interface (col4 lines 55). Hunter thus provides the motivation of using interfaces in order to effectively move packets between elements.

Basso discloses a device that stores search key in a register in the interface circuit (transferring said search key from said tree search engine to one or more key registers in an interface unit, page 3 paragraph [0048]).

It would have thus been obvious to a person skilled in the art to incorporate an interface comprising a key register disclosed by Basso with the method and system for testing filter rules using a cache containing a CAM and associating a search key with a particular thread number disclosed by Corl, Murase, Hunter, and Basso to effectively move packets between elements (tree search engine and CAM).

**Regarding claim 19**, the combination of Corl, Murase, Hunter, and Basso discloses all the limitations of claim 18, and Hunter further discloses a step in which the key associated with the first entry in the hash bin is requested by outputting (transferring) its address (particular address used to decode said particular thread number associated with said search key) to the memory (to

second register, col8 lines 47-49). This address generation thread is then said to further continue with another step (col8 lines 59-60). Hunter is however silent in referring to an interface unit, but with regards to the arguments in claim 18, it would have been obvious to a person skilled in the art to incorporate a second register in an interface unit to hold said particular addresses to effectively move packets between the elements of the tree search engine and CAM.

**Regarding claim 20**, the combination of Corl, Murase, Hunter, and Basso discloses all the limitations of claim 19, and Murase further discloses that the cache table (comprising a CAM) is comprised of a search key register (page 4 [0063]). It is thus obvious to a person skilled in the art that in order for the key register to store keys and search for matches in the cache, transferring of said search key from one of said search key registers in the interface to the cache (CAM) is necessary.

**Regarding claim 21**, the combination of Corl, Murase, Hunter, and Basso discloses all the limitations of claim 20, and Corl further discloses the method comprise searching a plurality of stored keys in the cache for the key (identifying a particular entry number in CAM, col3 lines 63-64). It is further stated that if there is a match (based on search key, whether said search key identifies and matches a particular entry in said CAM), a step in obtaining the stored action(s) corresponding to the key will be taken (identifying said data structure associated

with said packet of data based on said particular entry number in said CAM, col8 lines 34).

**Regarding claim 22**, the combination of Corl, Murase, Hunter, and Basso disclose all the limitations as discussed in the rejection of claim 21.

**Regarding claim 23**, the combination of Corl, Murase, Hunter, and Basso disclose all the limitations as discussed in the rejection of claim 21.

**Regarding claim 24**, the combination of Corl, Murase, Hunter, and Basso disclose all the limitations as discussed in the rejection of claim 23, and Murase further discloses that a cache table (in said CAM) is searched with a key and if any entry hit is found, the information that a hit entry (search key matches particular entry) was found is notified to the packet processing circuit along with the address of the cache table holding the hit entry (returns a particular entry number, page 3 [0055]).

**Regarding claim 25**, the combination of Corl, Murase, Hunter, and Basso disclose all the limitations as discussed in the rejection of claim 24, and Hunter further discloses that a second entry may quickly be determined by simply adding the forwarding database entry length (adding particular entry number to a base address, Examiner believes determination of forwarding database entry length be

comprise of determining starting address) to the address of the previous entry (shifting a particular entry number, col7 lines 7-9). Hunter is however silent in storing a result of said particular entry number added to said base address in result register, and reading said result in said result register.

Murase however, further discloses a hit record database (result register), which is updated when a hit occurs in the cache table (CAM, page 2 [0031]). Murase further discloses a status table be located in the hit record database (figure 4), which stores in each entry, the number of that entry (indexed in said result register based on thread number), the hit address in the cache table (particular entry number in CAM), the hit address in the forwarding table (result and pointer used to index into an appropriate entry in said data structure memory to identify said data structure associated with said packet of data, page4 paragraph [0067]). It would have thus been obvious to a person skilled in the art that this status table also store the result of said particular entry number added to said base address disclosed by Hunter with the hit record database disclosed by Murase, as this is simply the means in finding an address in forwarding table (appropriate entry in said data structure memory to identify said data structure associated with said packet of data) and storing such results in said result register.

**Regarding claim 26**, the combination of Corl, Murase, Hunter, and Basso disclose all the limitations as discussed in the rejection of claim 25.

**Regarding claim 28**, the combination of Corl, Murase, Hunter, and Basso disclose all the limitations as discussed in the rejection of claim 25, Hunter further discloses that each forwarding database entry has the same general format (col10 lines 17-20). It would have thus been obvious to a person skilled in the art to have to have the pointer in a same format when said tree is used to point to a forwarding database.

**Regarding claim 29**, the combination of Corl, Murase, Hunter, and Basso disclose all the limitations as discussed in the rejection of claim 19 and Hunter further discloses a retrieval, which includes causing a pipeline memory (one or more search key registers and said second register organized in pipeline manner) to access memory locations in an order that minimizes a worst case search of the forwarding database (access done by having a search key associated with one or more thread number/table number pairs, col3 lines 9-10).

---

### ***Conclusion***

8. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

a) Angle et al. (U.S 6,209,020), Distributed Pipeline Memory Architecture For A Computer System With Even and Odd Pids.

Art Unit: 2663

b) Spinney (U.S 5,414,704), Address Lookup In Packet Data Communication Link, Using Hashing and Content-Addressable Memory.

c) Michels et al. (U.S 6,453,358), Network Switching Device With Concurrent Key Lookups.

d) Sato et al. (U.S 2002/0038379), Routing Apparatus.

e) Sarkissian et al. (U.S 6,771,646), Associative Cache Structure For Lookups And Updates of Flow Records In A Network Monitor.

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Nguyen Ngo whose telephone number is (571) 272 - 8398. The examiner can normally be reached on Monday-Friday 7am - 3:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ricky Ngo can be reached on (571) 272-3139. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

**RICKY NGO  
PRIMARY EXAMINER**

Art Unit: 2663

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

\*\*\*

**Nguyen Ngo**

United States Patent & Trademark Office  
Patent Examiner AU 2663  
(571) 272-8398

*Ricky Ngo*  
RICKY NGO  
PRIMARY EXAMINER  
*4/28/05*